UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/693,302	10/24/2003	Balaji Rathakrishnan	MSFT-2848/306822.1	1494

23377 7590 11/30/2005

WOODCOCK WASHBURN LLP
ONE LIBERTY PLACE, 46TH FLOOR
1650 MARKET STREET
PHILADELPHIA, PA 19103

EXAMINER	
ORTIZ, BELIX M	
ART UNIT	**PAPER NUMBER**
2164	

DATE MAILED: 11/30/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/693,302	RATHAKRISHNAN ET AL.	
	Examiner	Art Unit	
	Belix M. Ortiz	2164	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *12/1/03*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. The abstract of the disclosure is objected to because, of the following reason:

   The abstract consists of more than 150 words.

2. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

   (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-6, 8-13, and 15-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krishnaprasad et al. (U.S patent 6,564,203) in view of Haas et al. (U.S. patent 6,892,204).

As to claim 1, Krishnaprasad et al. teaches a method of updating values in a complex structured type column in a relational database system (see abstract and column 1, lines 15-20), comprising the steps of:

representing modifications to values in the complex structured type column using a data structure that aggregates changes to the values at any level of a hierarchy of the complex structured column (see column 7, lines 27-37).

Krishnaprasad et al. does not teach computing the data structure in response to a data modification statement on the database to determine which values within the complex structured type column to update with the aggregated changes.

Haas et al. teaches spatially integrated relational database model with dynamic segmentation (see abstract), in which he teaches computing the data structure in response to a data modification statement on the database to determine which values within the complex structured type column to update with the aggregated changes (see column 39, lines 26-37).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Krishnaprasad et al. by the teaching of Davis et al., because computing the data structure in response to a data modification statement on the database to determine which values within the complex structured type column to update with the aggregated changes, would enable a method "In an alternative embodiment, each entry in the Derivations table 1170 defines how to calculate the value of a derived Attribute, a

calculation that involves other Attribute values. The entries in the Attribute Derivations table 1780 list the Attributes required by each calculation in the Derivations table 1770. The required Attributes, in turn, are used during data maintenance to (a) identify when a derived, instantiated Attribute must be updated (because one of the Attributes required by a derivation was changed) and (b) generate the SQL statement that accumulates all of the Attribute values together in order to compute a derived value. This table consists of the following two columns", (see Haas et al., column 39, lines 26-37).

As to claim 2, Krishnaprasad et al. as modified teaches a method comprising the further step of simultaneously updating multiple scalar values at different levels within the hierarchy of the complex structured type column (see Krishnaprasad et al., abstract; figure 1; and column 3, lines 32-42).

As to claim 3, Krishnaprasad et al. as modified teaches a method comprising the further step of simultaneously updating a scalar value in a table along with a complex structured type value in a complex structured type column of said relational database system (see Krishnaprasad et al., abstract; figure 1; and column 3, lines 32-42).

As to claim 4, Krishnaprasad et al. as modified teaches a method comprising the further step of embedding an INSERT/UPDATE/DELETE

statement inside a SET clause of an UPDATE statement (see Krishnaprasad et al., column 4, lines 36-44 and column 4, lines 48-57).

As to claim 5, Krishnaprasad et al. as modified teaches a method comprising the further step of embedding a plurality of nested SET clauses inside an outer-most UPDATE statement corresponding to each layer within the hierarchy of the complex structured type column (see Krishnaprasad et al., figure 1; column 2, lines 28-33; column 5, lines 14-16; and column 6, lines 4-12).

As to claim 6, Krishnaprasad et al. as modified teaches wherein the computing step comprises the steps of updating only indexes affected by specific scalar fields modified at various nesting levels by the SET clause in the UPDATE statement and updating only those rows of the index that correspond to the actual values that are modified by the UPDATE statement (see Krishnaprasad et al., abstract; claim 5; and column 2, lines 40-48).

As to claim 8, Krishnaprasad et al. teaches a relational database system responsive to database modification statements to store and update values in at least one complex structured type column (see abstract and column 1, lines 15-20), comprising:

a parser that parses a database modification statement and produces a description of changes to the database proposed by the database modification statement (see column 3, lines 66-67 and column 4, lines 1-7); and

a query optimizer that produces an execution algorithm to implement the database modification statement (see claim 9 and column 6, lines 4-12).

Krishnaprasad et al. does not teach a query execution engine that uses the execution algorithm to compute a data structure of the database modification statement to determine which values within a complex structured type column are to be updated, wherein the data structure represents values in the complex structured type column as an aggregation of changes to the values at any level of hierarchy of the complex structured type column, and said query execution engine applies the changes to the values in the complex structured type column that are to be updated.

Haas et al. teaches spatially integrated relational database model with dynamic segmentation (see abstract) in which he teaches a query execution engine that uses the execution algorithm to compute a data structure of the database modification statement to determine which values within a complex structured type column are to be updated, wherein the data structure represents values in the complex structured type column as an aggregation of changes to the values at any level of hierarchy of the complex structured type column, and said query execution engine applies the changes to the values in the complex structured type column that are to be updated (see column 39, 26-37).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Krishnaprasad et al. by the teaching of Haas et al., because a query execution engine that uses the execution algorithm to compute a data structure of the database modification statement to determine which values within a complex structured type column are to be updated, wherein the data structure represents values in the complex structured type column as an aggregation of changes to the values at any level of hierarchy of the complex structured type column, and said query execution engine applies the changes to the values in the complex structured type column that are to be updated, would enable a method "In an alternative embodiment, each entry in the Derivations table 1170 defines how to calculate the value of a derived Attribute, a calculation that involves other Attribute values. The entries in the Attribute Derivations table 1780 list the Attributes required by each calculation in the Derivations table 1770. The required Attributes, in turn, are used during data maintenance to (a) identify when a derived, instantiated Attribute must be updated (because one of the Attributes required by a derivation was changed) and (b) generate the SQL statement that accumulates all of the Attribute values together in order to compute a derived value. This table consists of the following two columns", (see Haas et al., column 39, lines 26-37).

As to claim 9, Krishnaprasad et al. as modified teaches wherein the query execution engine simultaneously updating multiple scalar values at different levels within the hierarchy of the complex structured type column (see Krishnaprasad et al., abstract; figure 1; and column 3, lines 32-42).

As to claim 10, Krishnaprasad et al. as modified teaches wherein the query execution engine simultaneously updating a scalar value in a table along with a complex structured type value in a complex structured type column of said relational database system (see Krishnaprasad et al., abstract; figure 1; and column 3, lines 32-42).

As to claims 11 and 19, Krishnaprasad et al. as modified teaches wherein the parser parses a SET clause of an UPDATE statement (see Krishnaprasad et al., column 4, lines 36-44 and column 4, lines 48-57).

As to claim 12, Krishnaprasad et al. as modified teaches wherein the parser parses the UPDATE statement in a plurality of nested SET clauses inside an outermost UPDATE statement corresponding to each level within the hierarchy of the complex structured type column (see Krishnaprasad et al., figure 1; column 2, lines 28-33; column 5, lines 14-16; and column 6, lines 4-12).

As to claims 13 and 21, Krishnaprasad et al. as modified teaches wherein the query execution engine updating only indexes affected by specific scalar fields modified at various nesting levels by the SET clause in the UPDATE statement and updating only those rows of the index that correspond to the actual values that are modified by the UPDATE statement (see Krishnaprasad et al., abstract; claim 5; and column 2, lines 40-48).

As to claim 15, Krishnaprasad et al. teaches a method of updating values in a collection-valued column in a relational database system (see abstract and column 1, lines 15-20), comprising the steps of:

representing modifications to values in the collection-valued column using a data structure that aggregates changes to the values at any level of a hierarchy of the collection-valued column (see column 7, lines 27-37).

Krishnaprasad et al. does not teach computing the data structure in response to a data modification statement on the database to determine which values within the collection-valued column to update with the aggregated changes.

Haas et al. teaches spatially integrated relational database model with dynamic segmentation (see abstract) in which he teaches computing the data structure in response to a data modification statement on the database to determine which values within the collection-valued column to update with the aggregated changes (see column 2, lines 40-67 and column 3, lines 1-9).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Krishnaprasad et al. by the teaching of Haas et al., because computing the data structure in response to a data modification statement on the database to determine which values within the collection-valued column to update with the aggregated changes, would enable a method "In an alternative embodiment, each entry in the Derivations table 1170 defines how to calculate the value of a derived Attribute, a calculation that involves other Attribute values. The entries in the Attribute Derivations table 1780 list the Attributes required by each calculation in the Derivations table 1770. The required Attributes, in turn, are used during data maintenance to (a) identify when a derived, instantiated Attribute must be updated (because one of the Attributes required by a derivation was changed) and (b) generate the SQL statement that accumulates all of the Attribute values together in order to compute a derived value. This table consists of the following two columns", (see Haas et al., column 39, lines 26-37).

As to claim 16, Krishnaprasad et al. as modified teaches a method comprising the further step of simultaneously updating multiple scalar values at different levels within the hierarchy of the collection-valued column (see Krishnaprasad et al., abstract; figure 1; and column 3, lines 32-42).

As to claim 17, Krishnaprasad et al. as modified teaches a method comprising the further step of simultaneously updating a scalar value in a table along with a value in a collection-valued column of said relational database system (see Krishnaprasad et al., abstract; figure 1; and column 3, lines 32-42).

As to claim 18, Krishnaprasad et al. teaches a relational database system responsive to database modification statements to store and update values in at least one collection-valued column (see abstract and column 1, lines 15-20), comprising:

a parser that parses a database modification statement and produces a description of changes to the database proposed by the database modification statement (see column 3, lines 66-67 and column 4, lines 1-7); and

a query optimizer that produces an execution algorithm to implement the database modification statement (see claim 9 and column 6, lines 4-12).

Krishnaprasad et al. does not teach a query execution engine that uses the execution algorithm to compute a data structure of the database modification statement to determine which values within a collection-valued column are to be updated, wherein the data structure represents values in the collection-valued column as an aggregation of changes to the values at any level of hierarchy of the complex structured type column, and said query execution engine applies the changes to the values in the collection-valued column that are to be updated.

Haas et al. teaches spatially integrated relational database model with dynamic segmentation (see abstract) in which he teaches a query execution engine that uses the execution algorithm to compute a data structure of the database modification statement to determine which values within a collection-valued column are to be updated, wherein the data structure represents values in the collection-valued column as an aggregation of changes to the values at any level of hierarchy of the collection-valued column, and said query execution engine applies the changes to the values in the collection-valued column that are to be updated (see column 2, lines 40-67 and column 3, lines 1-9).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Krishnaprasad et al. by the teaching of Haas et al., because a query execution engine that uses the execution algorithm to compute a data structure of the database modification statement to determine which values within a collection-valued column are to be updated, wherein the data structure represents values in the collection-valued column as an aggregation of changes to the values at any level of hierarchy of the complex structured type column, and said query execution engine applies the changes to the values in the collection-valued column that are to be updated, would enable a method "In an alternative embodiment, each entry in the Derivations table 1170 defines how to calculate the value of a derived Attribute, a calculation that involves other Attribute values. The entries in the Attribute Derivations table 1780 list the Attributes required by each calculation in the

Derivations table 1770. The required Attributes, in turn, are used during data maintenance to (a) identify when a derived, instantiated Attribute must be updated (because one of the Attributes required by a derivation was changed) and (b) generate the SQL statement that accumulates all of the Attribute values together in order to compute a derived value. This table consists of the following two columns", (see Haas et al., column 39, lines 26-37).

As to claim 20, Krishnaprasad et al. as modified teaches wherein the parser parses the UPDATE statement in a plurality of nested SET clauses inside an outermost UPDATE statement corresponding to each level within the collection-valued column (see Krishnaprasad et al., figure 1; column 2, lines 28-33; column 5, lines 14-16; and column 6, lines 4-12).

5. Claims 7, 14, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krishnaprasad et al. (U.S patent 6,564,203) in view of Haas et al. (U.S. patent 6,892,204) as applied to claims 1-6, 8-13, 15-21 above, and further in view of Graefe et al. (U.S. patent 6,122,644).

As to claims 7 and 14, Krishnaprasad et al. as modified does not teach a method comprising the further step/wherein the query execution engine applying the aggregated changes to the complex structured type column, wherein the

applying step is separate from the computing step so as to provide Halloween Protection.

Graefe et al. teaches system for Halloween protection in a database system (see abstract) in which he teaches a method comprising the further step of applying the aggregated changes to the complex structured type column, wherein the applying step is separate from the computing step so as to provide Halloween Protection (see figure 6 and column 5, lines 13-15).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Krishnaprasad et al. by the teaching of Graefe et al., because a method comprising the further step of applying the aggregated changes to the complex structured type column, wherein the applying step is separate from the computing step so as to provide Halloween Protection, would enable a method because, "Halloween protection is provided in the update plan by considering various other operators in addition to the eager spool operator and at locations anywhere in an update plan not just on the input side of an update operator. The Halloween protection system of the present invention provides improved update efficiency by interleaving the search and update phases to the full extent possible while still maintaining the required semantics, i.e., the search then update semantics of set-at-a-time pipelining, in the update plan", (see Graefe et al., column 3, lines 16-25).

As to claim 22, Krishnaprasad et al. as modified does not teach a method comprising the further step/wherein the query execution engine applying the aggregated changes to the collection-valued column, wherein the applying step is separate from the computing step so as to provide Halloween Protection.

Graefe et al. teaches system for Halloween protection in a database system (see abstract) in which he teaches a method comprising the further step of applying the aggregated changes to the collection-valued column, wherein the applying step is separate from the computing step so as to provide Halloween Protection (see figure 6 and column 5, lines 13-15).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Krishnaprasad et al. by the teaching of Graefe et al., because a method comprising the further step of applying the aggregated changes to the collection-valued column, wherein the applying step is separate from the computing step so as to provide Halloween Protection, would enable a method because, "Halloween protection is provided in the update plan by considering various other operators in addition to the eager spool operator and at locations anywhere in an update plan not just on the input side of an update operator. The Halloween protection system of the present invention provides improved update efficiency by interleaving the search and update phases to the full extent possible while still maintaining the required semantics, i.e., the search then update semantics of set-at-a-time pipelining, in the update plan", (see Graefe et al., column 3, lines 16-25).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Belix M. Ortiz whose telephone number is 571-272-4081. The examiner can normally be reached on moday-friday 9am-5pm.

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

C Rones

CHARLES RONES
SUPERVISORY PATENT EXAMINER

bmo

November 3, 2005